IN THE CLAIMS

1. (Currently Amended) A processor comprising:

a protected execution unit including a first execution unit and a second execution unit, the first execution unit and the second execution unit to process instructions in lock step;

a check unit coupled to the first execution unit and the second execution unit, the check unit to detect an error associated with processed instructions; and

at least one replay queue coupled to the check unit and to the protected execution unit, the at least one replay queue to issue a plurality of instructions to the protected execution unit for processing, to track the plurality of instructions issued to the protected execution unit, and to selectively reissue one or more of the plurality of instructions to the protected execution unit for which the check unit detects an error.

2. (Original) The processor of claim 1, wherein instructions are flushed from the execution unit when the check unit indicates an error.

3. (Previously Presented) The processor of claim 1, wherein the replay queue includes first and second pointers to respectively indicate a next instruction to issue and a next instruction to retire.

4-5. (Cancelled)

6. (Previously Presented) The processor of claim 1, wherein the execution units operate in lock step when the processor is in a high reliability mode and the execution units operate independently when the processor is in a high performance mode.

7. (Original) The processor of claim 1, wherein the processor implements a recovery algorithm if an instruction that triggers a replay generates a mismatch when it is replayed.

8. (Currently Amended) A method for executing instructions with high reliability, comprising:

temporarily storing a plurality of instructions in a replay buffer;

issuing a plurality of the instructions stored in the replay buffer to a first execution unit and a second execution unit of a protected execution unit;

receiving a plurality of results from the first execution unit and the second execution unit in response to the plurality of the instructions issued from the replay buffer;

checking the plurality of results generated by the first execution unit and the second execution unit in response to the plurality of the instructions issued from the replay buffer; and

signalling the replay buffer to selectively reissue one or more instructions of the plurality of instructions issued to the first execution unit and the second execution unit of the protected execution unit for which an error is detected during the checking.

9. (Original) The method of claim 8, wherein issuing the instruction comprises:

staging the instruction to the protected execution unit; and

adjusting a first flag in the buffer to indicate the instruction has been issued.

10. (Original) The method of claim 8, wherein adjusting the first flag comprises setting a first pointer to indicate a buffer slot in which the issued instruction is stored.

11. (Original) The method of claim 10, further comprising setting a second pointer to indicate a buffer slot in which a next instruction to retire is stored.

12. (Cancelled)

13. (Original) The method of claim 8, further comprising retiring the instruction when no error is indicated.

14. (Original) The method of claim 13, wherein retiring the instruction comprises:

adjusting a second pointer to indicate the instruction has retired; and

updating an architectural state data with the result generated by the instruction.

15. (Currently Amended) A computer system comprising:

a processor that includes:

a protected execution unit to execute instructions in a manner that facilitates ~~faciliates~~ soft error detection;

a check unit to monitor the protected execution unit and to generate a signal when an error is indicated;

a replay unit coupled to the protected execution unit and the check unit, the replay unit to temporarily store a plurality of instructions and provide the plurality of instructions to the protected execution unit for execution, the replay unit to track the plurality of instructions until they are retired, and the replay unit to repetitively replay selected instructions of the plurality of instructions for which the check unit indicates an error; and

a storage structure coupled to the processor, the storage structure to store a recovery algorithm, the storage structure to provide the recovery algorithm to the processor when a specified number of replays of the selected instructions does not eliminate the mismatch.

16. (Previously Presented) The computer system of claim 15, wherein the replay unit includes a queue to temporarily store the plurality of instructions with first and second pointers to indicate a next instruction to issue and a next instruction to retire, respectively.

17. (Previously Presented) The computer system of claim 16, wherein one or more execution units of the protected exeucution unit are flushed prior to the replay of selected instructions when an error is indicated.

18. (Cancelled)

19. (Original) The computer system of claim 16, wherein the storage structure is a non-volatile memory structure.

20. (Original) The computer system of claim 15, wherein the protected execution unit comprises first and second execution units and the replay unit provides identical instructions to the first and second execution units.

21. (Currently Amended) A processor comprising:

first and second execution cores to process identical instructions in lock step, each execution core having a fetch stage, a decode stage, a register stage, an execute stage, a detect stage, and a retirement stage in an instruction execution pipeline, the decode stage of each execution core including a replay unit to track a plurality of instructions that have yet to retire in the retirement stage; and

a check unit coupled to the first and second execution cores and the replay unit, the check unit to compare instruction results generated by the execution cores in their respective detect stages prior to retirement and to trigger the replay unit to resteer the first and second execution cores to re-execute one or more selected instructions in their respective instruction execution pipelines for which the instruction results generate a mismatch.

22. (Previously Presented) The processor of claim 21, wherein each replay unit includes

a buffer with a plurality of slots to store a plurality of instructions for execution, and

first and second pointers to indicate a next instruction to issue and a next instruction to retire, respectively.

23-24. (Cancelled)

25. (Previously Presented) A processor comprising:

an execution unit including

a first parity-protected storage structure having a first parity bit and

a second parity-protected storage structure having a second parity bit;

a check unit coupled to the first parity-protected storage structure and the second parity-protected storage structure, the check unit to monitor the the first parity-protected

storage structure and the second parity-protected storage structure, to detect a parity error in data accessed from the first parity-protected storage structure or in data accessed from the second parity-protected storage structure, and to signal a parity error; and

a replay queue coupled to the check unit and the execution unit, the at least one replay queue to issue a plurality of instructions to the protected execution unit for processing, to track the plurality of instructions issued to the protected execution unit, and to selectively reissue one or more of the plurality of instructions to the protected execution unit in response to the check unit detecting and signaling a parity error.

26. (Previously Presented) The processor of claim 25, wherein

a parity error indicates a soft error corrupted data after it was stored into the first parity-protected storage structure or the second parity-protected storage structure.

27. (Previously Presented) The processor of claim 25, wherein

the first parity-protected storage structure is a register file to store a first plurality of data blocks each of which has a first parity bit, and

the second parity-protected storage structure is a cache to store a second plurality of data blocks each of which has a second parity bit.

28. (Previously Presented) A processor comprising:

an execution unit including

a first protected storage structure having a first plurality of error correction control bits;

a second protected storage structure having a second plurality of error correction control bits;

a check unit coupled to the first parity-protected storage structure and the second parity-protected storage structure, the check unit to monitor the first protected storage structure and the second protected storage structure, to detect an error in data accessed from the first protected storage structure or in data accessed from the second protected storage structure, and to correct the error in the accessed data in response to the first or second plurality of error correction control bits, respectively; and

a replay queue coupled to the check unit and the execution unit, the at least one replay queue to issue a plurality of instructions to the protected execution unit for processing, to track the plurality of instructions issued to the protected execution unit, and to selectively reissue one or more of the plurality of instructions to the protected execution unit in response to the check unit detecting an error in instruction execution.

29. (Previously Presented) The processor of claim 28, wherein

the first protected storage structure is a register file to store a first plurality of data blocks each of which has a first plurality of error correction bits, and

the second protected storage structure is a cache to store a second plurality of data blocks each of which has a second plurality of error correction bits.

30. (Currently Amended) A check unit comprising:

a plurality of comparitors, one comparitor for each corresponding pair of execution units in a first execution core and a second execution core to compare execution results;

an OR gate having inputs coupled to the outputs of the plurality of comparitors, the OR gate to generate a logic value one at its output when any one of the plurality of comparitors indicate that their corresponding execution results do not match; and

wherein the output of the OR gate indicates an error when the check unit is enabled, wherein the output indicates the error to trigger a selective re-execution of one or more instructions for which the error is detected.

31. (Previously Presented) The check unit of claim 30, further comprising:

an AND gate having a first input coupled to an enable signal and a second input coupled to the output of the OR gate, the AND gate to enable the OR gate to generate an error signal and indicate an error in response to the enable signal.

32. (Previously Presented) The check unit of claim 30, further comprising:

a counter having an input coupled to the output of the OR gate, the counter to track a number of replays triggered by an instruction.

33. (Previously Presented) The check unit of claim 32, wherein

if the number of replays triggered by an instruction is equal to a specified number, the counter to invoke a recovery routine.